

# Trabalho1 :- Busy\_Police\_em\_Prolog.

autor(víctor\_colombo).  
ra(727356).

# O Problema

- O policial deve encontrar uma forma de chegar até o fugitivo.
- Ele deve contornar obstáculos: Pular carrinhos e evitar atravessar as paredes de borda.
- Como fazer isso em Prolog?

# O Cenário

- Cenário definido programaticamente: posições do policial, das escadas e dos carrinhos definidas no código.
- Ladrão no último andar; Policial no primeiro.
- Carrinhos devem ser pulados; escadas escaladas.

# Recursão

- Caso base: Prender o suspeito.
- `consegue(estado(Hor, Ver), estado(Hor, Ver), R, R)`.

# Pulando um Carrinho

- Caso haja um carrinho, não podemos colocar o policial na posição.
- Devemos mover o policial para a posição adjacente.
- Limitações.

# Subindo ou Descendo a escada

- Só podemos subir uma escada se estamos na mesma posição que ela.
- Só podemos descer a escada se estamos na mesma Horizontal e um andar acima dela.

# Andando Horizontalmente

- Podemos nos mover livremente pela horizontal.
- Exceto se houver um carrinho ou uma parede.

# Evitando Loops

- Manter em cada movimento um histórico dos estados visitados pelo PM.
- Evitar loops.
- Dar o caminho como resposta.



# Ordem da Recursão

- Captura.
- Escadas.
- Carrinho.
- Horizontal.

# CENÁRIO INTERESSANTE

FUGITIVO									
					CARRINHO	CARRINHO			ESCALADA
	ESCALADA	CARRINHO	ESCALADA					ESCALADA	
	ESCALADA	CARRINHO	CARRINHO						
PM	ESCALADA								

# AMBIENTE

- escada(2,1). escada(2,2). escada(2, 3).  
escada(4, 3). escada(9,3). escada(10, 4).
- Carrinho(3,2). Carrinho(4,2). Carrinho(3,3).  
Carrinho(6,4). Carrinho(7,4).

?- consigue(estado(1,1), estado(1,5), [], R).

- True.
- R = ...;
- False.

# EXTRA EXTRA!

- Precisamos de evidências!!!
- Coletar as evidências para prender o fugitivo.
- Armazenar as evidências numa lista.
- Verificar se a lista possui todas as evidências ao tentar efetuar a prisão.